



## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **06086272 A**(43) Date of publication of application: **25 . 03 . 94**

(51) Int. Cl.

**H04N 7/137****G06F 15/70**(21) Application number: **04253538**(22) Date of filing: **31 . 08 . 92**(71) Applicant: **SONY CORP**(72) Inventor: **HASHINO TSUKASA  
KAWAGUCHI KUNIO**(54) **MOVING VECTOR DETECTOR**

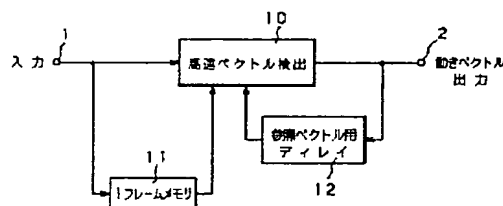
## (57) Abstract:

**PURPOSE:** To shorten an arithmetic time by successively comparing the middle calculated results of plural evaluation functions, and moving to the calculation of the new evaluation function between the pre-block and a block to be processed.

**CONSTITUTION:** An input television picture signal is transmitted to a high speed vector detecting circuit 10 and a one frame memory 11. The circuit 10 successively compares an in-block residual absolute value sum as the evaluation function calculated between a reference block at a corresponding spatial position and a block T with the cumulative middle result of the in-block residual absolute value at the time of searching the in-block residual absolute value sum being the evaluation function calculated between the block T and the arbitrary pre-block in a prescribed inspection space I, by referring to a moving vector held in a delay 12 for a reference vector. Then, when the cumulative middle result is more than the in-block residual absolute value sum, the calculation of the in-block residual absolute value sum is stopped, and the calculation of the new in-block residual absolute value sum between the next

pre-block in the prescribed inspection space I and the block T is started.

COPYRIGHT: (C)1994,JPO&Japio



.(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-86272

(43)公開日 平成6年(1994)3月25日

(51)Int.Cl. <sup>5</sup>	識別記号	庁内整理番号	F I	技術表示箇所
H 0 4 N 7/137	Z			
G 0 6 F 15/70	4 1 0	8837-5L		

審査請求 未請求 請求項の数4(全 8 頁)

(21)出願番号 特願平4-253538

(22)出願日 平成4年(1992)8月31日

(71)出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72)発明者 橋野 司

東京都品川区北品川6丁目7番35号 ソニー株式会社内

(72)発明者 川口 邦雄

東京都品川区北品川6丁目7番35号 ソニー株式会社内

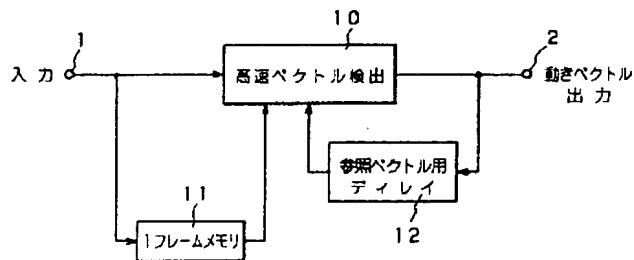
(74)代理人 弁理士 小池 晃 (外2名)

(54)【発明の名称】 動きベクトル検出装置

(57)【要約】

【構成】 処理対象ブロックTの時空間的近傍ブロックで求められている動きベクトルを保持する参照ベクトル用ディレイ12と、参照ブロックと処理対象ブロックTとの間で計算した第1の評価関数と処理対象ブロックTと前ブロックとの間で計算する第2の評価関数の累積途中結果とを逐次比較し、累積途中結果が第1の評価関数よりも大きくなった場合には第2の評価関数の計算を中止して新たな第2の評価関数計算に移行し、第2の評価関数が第1の評価関数より小さい場合には、第1の評価関数を第2の評価関数で更新する高速ベクトル検出回路10とを有する。

【効果】 演算時間の短縮、ハードウェアの小型化、高精度のベクトル検出ができる。



## 【特許請求の範囲】

【請求項1】 現画像内の複数画素群からなる処理対象ブロックと、当該現画像に対して時間的に前となる前画像の所定の探索空間内の複数画素群からなる前ブロックとの間で所定の評価関数を用いたブロックマッチングを行って動きベクトルの検出を行う動きベクトル検出装置において、

上記処理対象ブロックの近傍に存在する近傍ブロックで既に求められている動きベクトルを参照し、この参照動きベクトルに対応する空間位置の参照ブロックと上記処理対象ブロックとの間で計算した第1の評価関数と、上記処理対象ブロックと上記所定の探索空間内の任意の前ブロックとの間で計算する第2の評価関数の計算途中結果とを逐次比較し、上記第2の評価関数の計算途中結果が上記第1の評価関数よりも大きくなった場合には、上記第2の評価関数の計算を中止して上記所定の探索空間内の次の前ブロックと上記処理対象ブロックとの間の新たな第2の評価関数の計算に移行することを特徴とする動きベクトル検出装置。

【請求項2】 上記処理対象ブロックと上記所定の探索空間内の任意の前ブロックとの間で計算した第2の評価関数が上記第1の評価関数よりも小さくなった場合には、上記第1の評価関数を当該第2の評価関数で更新して、上記所定の探索空間内の次の前ブロックと上記処理対象ブロックとの間の新たな第2の評価関数の計算に移行することを特徴とする請求項1記載の動きベクトル検出装置。

【請求項3】 上記参照ブロックは、上記処理対象ブロックに対して時間的に近傍のブロックとすることを特徴とする請求項1及び2記載の動きベクトル検出装置。

【請求項4】 上記参照ブロックは、上記処理対象ブロックに対して空間的に近傍のブロックとすることを特徴とする請求項1及び2記載の動きベクトル検出装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】本発明は、例えばテレビジョン画像中の動きの方向及び量を示す動きベクトルを検出するための動きベクトル検出装置に関するものである。

## 【0002】

【従来の技術】例えばテレビジョン画像中の動きの方向

$$R(a, b) = \sum_{m_x=0}^{N_x-1} \sum_{m_y=0}^{N_y-1} |I(a, b)(m_x, m_y) - T(m_x, m_y)| \quad \dots (1)$$

【0006】ここで、数1の数式(1)の式中Iは上記探索空間を示し、式中Tは上記処理対象ブロックを、式中(a, b)はブロックの空間位置を表す(X, Y)座標を、式中(m<sub>x</sub>, m<sub>y</sub>)はブロック内の各画素の位置を表す(X, Y)座標を、式中R(a, b)は上記空間位置(a, b)での残差(差分)のブロック内絶対値和を示す。

## 【0007】

及び量を示す動きベクトルを検出する場合には、現画像内の複数画素からなる任意のブロックを処理対象とし、この処理対象ブロックが、当該現画像に対する時間的に前の画像(前画像)のどの位置に存在するかをいわゆるブロックマッチング法によって調べる方法が通常用いられている。

【0003】すなわち、図4に示すように、時間軸方向に隣接する2枚の画像があった場合を考える。この図4において、画像が動画像であればこれら画像はフレーム又はフィールドとなる。ここで、これら画像をフレームとすると、これら2枚のフレームのうちの一方のフレームは他方のフレームに対して時間的に前のフレーム(前フレーム)となり、また、上記他方のフレームは当該前フレームに対して時間的に後のフレーム(後フレーム)となる。したがって、上記ブロックマッチング法では、当該図4の上記後フレーム(すなわち現フレーム)の処理対象フレームTが、上記前フレームのどの位置に存在したかを調べることになる。

【0004】この時、最も一般的なブロックマッチング法による動きベクトル検出では、先ず、例えば図5のように、上記後フレーム(現フレーム)の例えばN<sub>x</sub>×N<sub>y</sub>画素からなる処理対象ブロックTと、当該後フレームの処理対象ブロックTと同一空間位置近傍でかつ当該処理対象ブロックTよりも大きい上記前フレームの探索空間I(M<sub>x</sub>×M<sub>y</sub>画素からなる空間)とを重ね合わせる。ここで、上記処理対象ブロックTをこの探索空間I内の探索範囲(横方向にM<sub>x</sub>-N<sub>x</sub>+1画素で、縦方向にM<sub>y</sub>-N<sub>y</sub>+1画素となる)上で動かし、所定の評価関数に基づいて上記処理対象ブロックTと上記探索空間I内の当該処理対象ブロックTに対応する大きさの前ブロックとの間で最もマッチングのとれている位置を求める。上記処理対象ブロックTの位置と、上記探索空間I内の上記最もマッチングのとれている上記前ブロックの位置との間の方向及び量(距離)を動きベクトルとして検出する。なお、当該所定の評価関数としては例えば数1の数式(1)に示すようなブロック内残差の絶対値和を用いることが多い。

## 【0005】

## 【数1】

【発明が解決しようとする課題】ところで、上記数式(1)を上記探索空間Iの探索範囲内の全ての空間位置(a, b)に適用した場合、この探索範囲が大きいと膨大な処理時間が必要になってしまう。すなわち、上記探索空間I内の全ての前ブロックと上記処理対象ブロックTとの間のブロックマッチングのための上記評価関数(ブロック内残差絶対値和)を求めるためには、膨大な処理時間が必要になる。

【0008】このようなことから、処理時間を短縮するために、例えば、初めから演算量の大きい高解像度画像（すなわち原画像）を用いて重ね合わせを行うようにしないで、例えば2×2画素の平均をとることによって低解像度の画像を作成し、更に、同様の方法を利用して順次低解像度の画像を作ってこれら画像を階層化し、解像度の低い画像から徐々に動きベクトルの範囲をしばり込むことにより、トータルの演算時間を短縮するいわゆるピラミッド構造を適用した方法を使用することがある。このピラミッド構造処理では、各階層の画像に対応できる大きな容量のメモリ手段が必要となる。その他、サンプリングによってブロック内のサンプル数を減らして演算量を少なくする方法等、多数あるが、これらの方法では、演算時間、ハードウェア規模、ベクトル検出精度の3つの条件を全て満足できない。

【0009】そこで本発明は、このような実情に鑑みてなされたものであり、演算時間が短く、ハードウェア規模も小さく、かつベクトル検出精度も高い動きベクトル検出が可能な動きベクトル検出装置の提供を目的とするものである。

【0010】

【課題を解決するための手段】本発明の動きベクトル検出装置は、上述の目的を達成するために提案されたものであり、現画像内の複数画素群からなる処理対象ブロックと、当該現画像に対して時間的に前となる前画像の所定の探索空間内の複数画素群からなる前ブロックとの間で所定の評価関数を用いたブロックマッチングを行って動きベクトルの検出を行う動きベクトル検出装置であって、上記処理対象ブロックの近傍に存在する近傍ブロックで既に求められている動きベクトルを参照し、この参照動きベクトルに対応する空間位置の参照ブロックと上記処理対象ブロックとの間で計算した第1の評価関数と、上記処理対象ブロックと上記所定の探索空間内の任意の前ブロックとの間で計算する第2の評価関数の計算途中結果とを逐次比較し、上記第2の評価関数の計算途中結果が上記第1の評価関数よりも大きくなった場合には、上記第2の評価関数の計算を中止して上記所定の探索空間内の次の前ブロックと上記処理対象ブロックとの間の新たな第2の評価関数の計算に移行するようにしたものである。

【0011】また、本発明の動きベクトル検出装置は、上記処理対象ブロックと上記所定の探索空間内の任意の前ブロックとの間で計算した第2の評価関数が上記第1の評価関数よりも小さくなった場合には、上記第1の評価関数を当該第2の評価関数で更新して、上記所定の探索空間内の次の前ブロックと上記処理対象ブロックとの間の新たな第2の評価関数の計算に移行するようにもしている。

【0012】すなわち、具体的に言うと、本発明の動きベクトル検出装置は、現画像内の複数画素群からなる処

理対象ブロックと、当該現画像に対して時間的に前となる前画像の所定の探索空間内の複数画素群からなる前ブロックとの間で求めたブロック内残差絶対値和を上記所定の評価関数として用いたブロックマッチングを行い、上記前ブロックから上記処理対象ブロックへの動きの方向及び量を示す動きベクトルの検出を行う動きベクトル検出装置であって、上記処理対象ブロックの近傍に存在する近傍ブロックで既に求められている動きベクトルを保持する動きベクトル保持手段と、この動きベクトル保持手段に保持された動きベクトルを参照し、この参照動きベクトルに対応する空間位置の参照ブロックと上記処理対象ブロックとの間で計算した第1の評価関数としての第1のブロック内残差絶対値和と、上記処理対象ブロックと上記所定の探索空間内の任意の前ブロックとの間で計算する第2の評価関数である第2のブロック内残差絶対値和を求めるためのブロック内残差絶対値の累積途中結果とを逐次比較し、上記第2の評価関数の累積途中結果が上記第1の評価関数の第1のブロック内残差絶対値和よりも大きくなった場合には上記第2の評価関数の計算を中止して上記所定の探索空間内の次の前ブロックと上記処理対象ブロックとの間の新たな第2の評価関数の計算に移行し、上記処理対象ブロックと上記所定の探索空間内の任意の前ブロックとの間で計算した第2の評価関数が上記第1の評価関数よりも小さくなった場合には上記第1の評価関数を当該第2の評価関数で更新して、上記所定の探索空間内の次の前ブロックと上記処理対象ブロックとの間の新たな第2の評価関数の計算に移行すると共に、この更新して得た第1の評価関数と、上記処理対象ブロックと上記次の前ブロックとの間で計算する新たな第2の評価関数の累積途中結果とを逐次比較する処理に移行する高速ベクトル検出手段とを有しているものである。

【0013】ここで、上記参照ブロックは、上記処理対象ブロックに対して時間的に近傍のブロックとすることができる。この時間的に近傍のブロックとしては、例えば、上記処理対象ブロックの属する画像の一つ前の画像のブロックでありかつ上記処理対象ブロックと空間的位置が同位置となるブロックを用いることができる。

【0014】また、上記参照ブロックは、上記処理対象ブロックに対して空間的に近傍のブロックとすることもできる。この空間的に近傍のブロックとしては、例えば、通常のテレビジョン信号の場合を考慮すると、上記処理対象ブロックの属する画像内の当該処理対象ブロックの縦方向（垂直方向）上隣のブロック（処理対象ブロックに対して1ブロック遅延されたブロック）や、上記処理対象ブロックの横方向（水平方向）の左隣のブロック（処理対象ブロックに対して1ラインブロック遅延サレタブロック）を用いることができる。

【0015】

【作用】本発明によれば、現画像の処理対象ブロックの

動きベクトルとこの処理対象ブロックに対する空間的近傍のブロックや前画像の同一空間位置のブロック（時間的近傍ブロック）の動きベクトルとは類似している場合が多いことを考慮し、所定の探索空間内の任意の前ブロックと処理対象ブロックとのブロックマッチングによる動きベクトルの検出を行う際に、処理対象ブロックの時空間的に近傍に存在する近傍ブロックで既に求められている参照動きベクトルに対応する参照ブロックと処理対象ブロックとの間で求めた第 1 の評価関数と、処理対象ブロックと探索空間内の任意の前ブロックとの間で計算する第 2 の評価関数の計算途中結果とを逐次比較し、第 2 の評価関数の計算途中結果が第 1 の評価関数よりも大きくなった場合には必要とする動きベクトルが得られないと判断して、その第 2 の評価関数の計算を中止して、次の前ブロックと処理対象ブロックとの間での新たな第 2 の評価関数の計算に移行することで、探索空間内の全ての前ブロックに対して第 2 の評価関数の全ての計算を行う必要がなくなり、演算時間の大幅な短縮を図ることができるようになる。

【0016】また、処理対象ブロックと探索空間内の前ブロックとの間で計算した第 2 の評価関数が第 1 の評価関数よりも小さくなった場合には、第 1 の評価関数を第 2 の評価関数で更新することにより、第 2 の評価関数が最も小さくなる探索空間内の前ブロックを検出することができるようになり、必要とする動きベクトルを検出することができるようになる。

【0017】

【実施例】以下、本発明の動きベクトル検出装置の実施例を図面を参照しながら説明する。なお、本発明における画像としては、フィールドを単位とする場合又はフレームを単位とする場合を考えることが、本実施例ではフィールドを単位とする場合について述べる。

【0018】本実施例の動きベクトル検出装置は、図 1 に示すように、現フレーム内の複数画素群からなる処理対象ブロック T と、当該現フレームに対して時間的に前となる前フレームの所定の探索空間 I 内の複数画素群からなる前ブロックとの間でブロック内残差絶対値和を所定の評価関数として用いたブロックマッチングを行い、上記前ブロックの位置から上記処理対象ブロック T の位置への動きの方向及び量を示す動きベクトルの検出を行う動きベクトル検出装置であって、一般的に上記現フレームの処理対象ブロック T の動きベクトルと当該処理対象ブロック T の空間的に近傍のブロックや前フレームの同一空間位置のブロックの動きベクトルとは類似している場合が多いことを考慮しており、上記処理対象ブロック T の時空間的に近傍に存在する近傍ブロックで既に求められている動きベクトルを保持する動きベクトル保持手段である参照ベクトル用ディレイ 12 と、以下に述べる高速ベクトル検出回路 10 とを有するものである。

【0019】上記高速ベクトル検出回路 10 は、上記参

照ベクトル用ディレイ 12 に保持された動きベクトルを参照し、この参照動きベクトルに対応する空間位置の参照ブロックと上記処理対象ブロック T との間で計算した第 1 の評価関数としての第 1 のブロック内残差絶対値和と、上記処理対象ブロック T と上記所定の探索空間 I 内の任意の前ブロックとの間で計算する第 2 の評価関数である第 2 のブロック内残差絶対値和を求める際のブロック内残差絶対値の累積途中結果とを逐次比較し、上記第 2 のブロック内残差絶対値和演算の累積途中結果が上記第 1 のブロック内残差絶対値和よりも大きくなった場合には、上記第 2 のブロック内残差絶対値和の計算を中止して上記所定の探索空間 I 内の次の前ブロックと上記処理対象ブロック T との間の新たな第 2 のブロック内残差絶対値和の計算に移行するような処理を行うものである。

【0020】さらに、この高速ベクトル検出回路 10 は、上記第 2 のブロック内残差絶対値和演算の累積途中結果が上記第 1 のブロック内残差絶対値和よりも大きくなった場合の処理と共に、上記処理対象ブロック T と上記所定の探索空間 I 内の任意の前ブロックとの間で計算した第 2 のブロック内残差絶対値和が上記第 1 のブロック内残差絶対値和よりも小さくなった場合には、上記第 1 のブロック内残差絶対値和を当該第 2 のブロック内残差絶対値和で更新して、上記所定の探索空間 I 内の次の前ブロックと上記処理対象ブロック T との間の新たな第 2 のブロック内残差絶対値和の計算に移行すると共に、上記更新して得た第 1 のブロック内残差絶対値和と、上記処理対象ブロック T と上記次の前ブロックとの間で計算する新たな第 2 のブロック内残差絶対値和の計算途中結果とを逐次比較する処理に移行する処理をも行うものである。

【0021】すなわち、この図 1 に示す本実施例の動きベクトル検出装置において、入力端子 1 には、例えばデジタルテレビジョン画像信号が供給される。このテレビジョン画像信号は、上記高速ベクトル検出回路 10 に送られると共に、1 フレーム分の画像データを記憶する 1 フレームメモリ 11 に送られる。上記高速ベクトル検出回路 10 は、上記参照ベクトル用ディレイ 12 からの既に求められている動きベクトル及び上記 1 フレームメモリ 11 内に記憶された 1 フレーム分の画像データのうちの上記探索空間 I の画像データを用いて、後述する図 2 のフローチャートに示すような手順によって高速動きベクトル検出処理を行う。また、上記高速ベクトル検出回路 10 で検出された動きベクトルは、出力端子 2 から出力される。

【0022】ここで、図 2 のフローチャートを用いて、上記高速動きベクトル検出処理の際の上記高速ベクトル検出回路 10 における処理の手順を説明する。

【0023】この図 2 のフローチャートにおいて、先ず、ステップ S 1 では、処理対象ブロック T の時空間的

に近傍に存在する近傍ブロックで過去に処理されて求められていると共に上記参照ベクトル用ディレイ12に記憶されている動きベクトルを参照し（参照動きベクトル $(V_x, V_y)$ ）、この参照動きベクトル $(V_x, V_y)$ に対応する空間位置（参照ブロック）と上記処理対象ブロックTとの間で上記第1の評価関数である第1のブロック内残差絶対値和 $R(V_x, V_y)$ を前述した数式(1)を用いて算出し、求めた第1のブロック内残差絶対値和 $R(V_x, V_y)$ と上記参照動きベクトルの位置を示す $(V_x, V_y)$ とを、以下の数式(2)～数式(4)のように $R_{min}$ 、 $V_{x_{min}}$ 、 $V_{y_{min}}$ として当該高速ベクトル検出回路10内のメモリに登録しておく。なお、上記参照動きベクトル $(V_x, V_y)$ と現在求めようとしている処理対象ブロックTの動きベクトルとが同一であれば、各探索空間Iのなかで上記第1のブロック内残差絶対値和 $R(V_x, V_y)$ が残差絶対値和の最小値をとることになる。

【0024】

$$R_{min} \equiv R(V_x, V_y) \quad \dots\dots\dots (2)$$

$$V_{x_{min}} \equiv V_x \quad \dots\dots\dots (3)$$

$$V_{y_{min}} \equiv V_y \quad \dots\dots\dots (4)$$

【0025】上記ステップS1の後、ステップS2に進む。このステップS2では、前述の数式(1)により、探索空間Iの探索範囲内の各位置（各前ブロック）における第2の評価関数である第2のブロック内残差絶対値和を求めるための積算を行う。

【0026】このステップS2の後、ステップS3に進む。このステップS3では、上記ステップS2における各空間位置(a,b)での第2のブロック内残差絶対値和を求めるための足し込み演算（積算）の際に、上記第2のブロック内残差絶対値和の演算の累積途中結果 $R(M_x, M_y)$ と、上記メモリ内に登録された第1のブロック内残差絶対値和 $R_{min}$ との比較を逐次行う。このステップS3で、上記第2のブロック内残差絶対値和の累積途中結果 $R(M_x, M_y)$ が上記登録されている第1のブロック内残差絶対値和 $R_{min}$ よりも大きくなったと判断した場合（Yes）には、上記空間位置(a,b)の前ブロックからは必要とする動きベクトルを求めることができないと判断して、当該第2のブロック内残差絶対値和の演算処理を中止して、ステップS7に進む。

【0027】このステップS7では、上記探索空間I内の全ての前ブロック（探索範囲の全空間位置）についての動きベクトル検出処理が終了したか否かの判断を行う。ここでは、上記探索空間I内の探索範囲の全空間位置についての動きベクトル検出処理が未だ終了していない（No）ため、上記ステップS2に戻る。

【0028】すなわち、このステップS2に戻ると言うことは、次の位置（次の前ブロック）としての例えば空間位置(a+1,b)の前ブロックに対する上記第2のブロック内残差絶対値和を求めるための足し込み処理（積算処理）に移ることを意味する。その後、再びステップS3

に進む。

【0029】また、上記ステップS3において、上記第2のブロック内残差絶対値和の累積途中結果 $R(M_x, M_y)$ が上記第1のブロック内残差絶対値和 $R_{min}$ よりも小さいと判断している間（No）は、ステップS4に進む。このステップS4では、処理対象ブロックT内の全画素と上記前ブロック内の全画素との間の上記第2のブロック内残差絶対値和を求めるための足し込み処理が終了したか否かの判断がなされ、終了していないと判断した場合（No）にはステップS2に戻る。

【0030】ここで、上記ステップS3での判断において、上記ステップS2で当該求められた第2のブロック内残差絶対値和の累積途中結果 $R(M_x, M_y)$ が、上記ステップS4において上記第2のブロック内残差絶対値和の演算が終了したと判断された場合（Yes）であってもなお上記第1のブロック内残差絶対値和 $R_{min}$ よりも小さいと判断された場合（Yes）は、ステップS5に進む。

【0031】このステップS5では、上記ステップS4において上記第2のブロック内残差絶対値和の演算が終了したと判断された場合の上記累積途中結果 $R(M_x, M_y)$ すなわち第2のブロック内残差絶対値和が、上記第1のブロック内残差絶対値和 $R_{min}$ よりも小さいか否かの判断がなされる。このステップS5において、上記第2のブロック内残差絶対値和（すなわち累積演算終了時の $R(M_x, M_y)$ ）が上記第1のブロック内残差絶対値和 $R_{min}$ よりも小さいと判断された場合（Yes）は、ステップS6に進む。

【0032】このステップS6では、数式(5)～数式(7)に示すように、上記ステップS5でYesと判断された時の上記第2のブロック内残差絶対値和（ $R(M_x, M_y)$ ）を、上記第1のブロック内残差絶対値和 $R_{min}$ として再登録（更新）すると共に、この時上記メモリに登録されていた参照動きベクトル $(V_x, V_y)$ の位置も当該第2のブロック内残差絶対値和に対応した動きベクトルの位置に再登録（更新）する。

【0033】

$$R_{min} \equiv R(M_x, M_y) \quad \dots\dots\dots (5)$$

$$V_{x_{min}} \equiv M_x \quad \dots\dots\dots (6)$$

$$V_{y_{min}} \equiv M_y \quad \dots\dots\dots (7)$$

【0034】その後、ステップS6の次のステップS7に進み、また、上記ステップS5においてNoと判断された場合にもステップS7に進む。

【0035】このステップS7では、上述したように、上記探索空間I内の探索範囲の全空間位置についての動きベクトル検出処理が終了したか否かの判断を行い、このステップS7で上記探索空間I内の探索範囲の全空間位置についての動きベクトル検出処理が終了したと判断した場合（Yes）には、最終的に上記メモリに登録されている最小となる動きベクトル $(V_{x_{min}}, V_{y_{min}})$ の

位置を上記処理対象ブロックTの動きベクトルとして取り出す。

【0036】なお、本実施例における上記参照ブロックとしては、上記処理対象ブロックTに対して時間的に近傍のブロックとすることができる。この時間的に近傍のブロックとしては、例えば、上記処理対象ブロックTの属するフレームの一つ前のフレーム（前フレーム）のブロックでありかつ上記処理対象ブロックTと空間的位置が同位置となるブロックを用いることができる。

【0037】また、上記参照ブロックは、上記処理対象ブロックTに対して空間的に近傍のブロックとすることもできる。この空間的に近傍のブロックとしては、例えば、通常のテレビジョン信号の場合を考慮すると、図3に示すように、上記処理対象ブロックTの属するフレーム内の当該処理対象ブロックTの縦方向（垂直方向）上隣のブロックや、上記処理対象ブロックTの横方向（水平方向）の左隣のブロックを用いることができる。

【0038】なお、上記空間的に近傍のブロックを上記上隣のブロックとする場合の上記図1の参照ベクトル用ディレイ12は上記処理対象ブロックTに対する1ブロック遅延を行うディレイ回路となり、上記空間的に近傍のブロックを上記左隣のブロックとする場合の上記図1の参照ベクトル用ディレイ12は上記処理対象ブロックTに対する1ラインブロック遅延を行うディレイ回路となる。

【0039】上述したように、本実施例の動きベクトル検出装置では、上記現フレームの処理対象ブロックTの動きベクトルとその近傍のブロックや前フレームの同一空間位置のブロック（時間的に近傍のブロック）の動きベクトルは類似している場合が多いことを考慮し、上記探索空間I内の任意の前ブロックと処理対象ブロックTとのブロックマッチングによる動きベクトルの検出を行う際に、上記処理対象ブロックTの時空間的に近傍に存在するブロックで既に求められている参照動きベクトルに対応する参照ブロックと当該処理対象ブロックTとの間で求めた第1の評価関数（第1のブロック内残差絶対値和）と、上記処理対象ブロックTと探索空間I内の任意の前ブロックとの間で計算する第2の評価関数（第2のブロック内残差絶対値和）の累積途中結果とを逐次比較し、上記第2のブロック内残差絶対値和の累積途中結果が上記第1のブロック内残差絶対値和よりも大きくなった場合には必要とする動きベクトルが得られないと判断して、その第2のブロック内残差絶対値和の累積計算を中止して、次の前ブロックと処理対象ブロックTとの間で新たな第2のブロック内残差絶対値和の計算に移行することで、探索空間I内の全ての前ブロックに対して第2のブロック内残差絶対値和の全ての計算を行う必要がなくなり、演算時間の大幅な短縮を図ることができるようになる。

【0040】また、本実施例によれば、処理対象ブロック

クTと探索空間I内の前ブロックとの間で計算した第2のブロック内残差絶対値和が第1のブロック内残差絶対値和よりも小さくなった場合には、第1のブロック内残差絶対値和を第2のブロック内残差絶対値和で更新することにより、第2のブロック内残差絶対値和が最も小さくなる探索空間I内の前ブロックを検出することができるようになり、これにより、通常のブロックマッチングを行った場合と同等のベクトル検出精度によって必要とする動きベクトルを検出することができるようになる。

【0041】このように、本実施例の動きベクトル検出装置は、前述したピラミッド構造処理のように比較的大きなメモリ量を必要とせず、通常のブロックマッチング法を行った場合と同等の動きベクトル検出精度を持ち、かつマッチング処理に要する時間を大幅に短縮することが可能となっている。

【0042】

【発明の効果】上述のように、本発明の動きベクトル検出装置においては、現画像の処理対象ブロックの動きベクトルと、この処理対象ブロックに対する空間的に近傍のブロックや前画像の同一空間位置のブロックの動きベクトルとは類似している場合が多いことを考慮し、処理対象ブロックの時空間的に近傍に存在するブロックで既に求められている参照動きベクトルに対応する参照ブロックと処理対象ブロックとの間で求めた第1の評価関数と、処理対象ブロックと探索空間内の任意の前ブロックとの間で計算する第2の評価関数の計算途中結果とを逐次比較し、第2の評価関数の計算途中結果が第1の評価関数よりも大きくなった場合には必要とする動きベクトルが得られないと判断して、その第2の評価関数の計算を中止して、次の前ブロックと処理対象ブロックとの間で新たな第2の評価関数の計算に移行することで、探索空間内の全ての前ブロックに対して第2の評価関数の全ての計算を行う必要がなくなり、演算時間の大幅な短縮を図ることができるようになる。

【0043】また、本発明においては、処理対象ブロックと探索空間内の前ブロックとの間で計算した第2の評価関数が第1の評価関数よりも小さくなった場合には、第1の評価関数を第2の評価関数で更新することにより、第2の評価関数が最も小さくなる探索空間内の前ブロックを検出することができるようになり、したがって、必要とする動きベクトルを検出することができるようになる。

【0044】すなわち、本発明の動きベクトル検出装置によれば、演算時間が短く、ハードウェア規模も小さく、またベクトル検出精度も高い動きベクトル検出が可能となる。

【図面の簡単な説明】

【図1】本実施例の動きベクトル検出装置の基本構成例を示すブロック回路図である。

【図2】本実施例装置での高速動きベクトル検出処理の

手順を示すフローチャートである。

・【図3】本実施例における空間的近接ブロックを示す図である。

【図4】ブロックマッチング処理の際の後フレーム及び処理対象ブロックと、前フレーム及び探索空間を説明するための図である。

【図5】ブロックマッチングの際の探索範囲を説明するための図である。

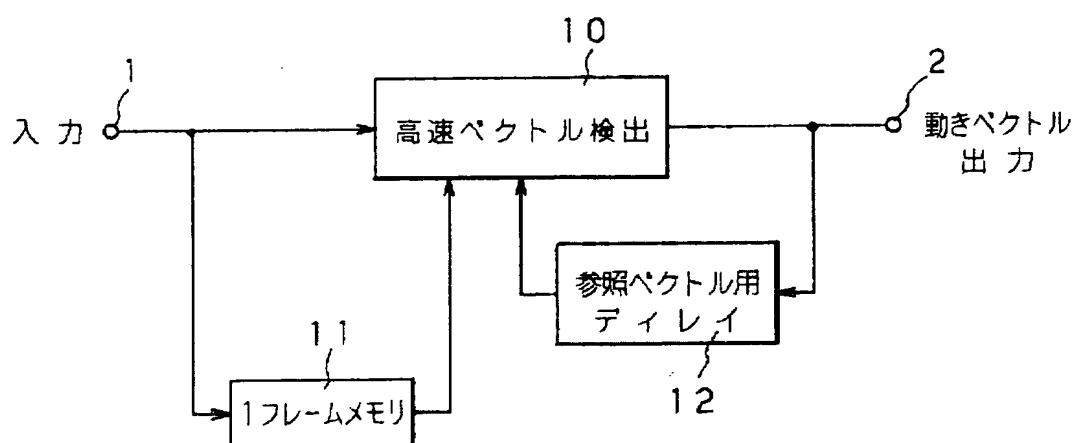
【符号の説明】

10 . . . . . 高速ベクトル検出回路

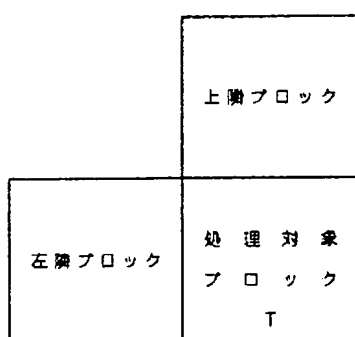
11 . . . . . 1フレームメモリ

12 . . . . . 参照ベクトル用ディレイ

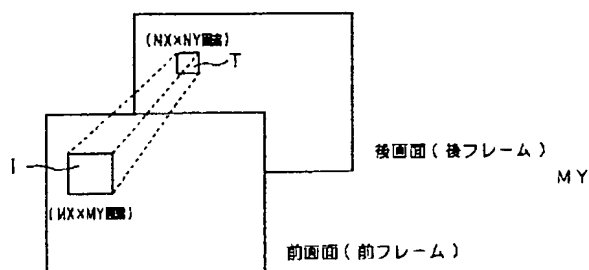
【図1】



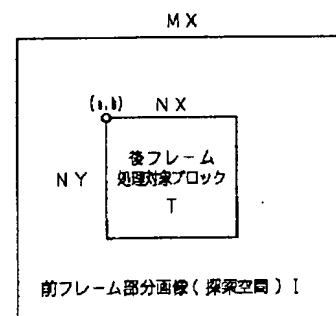
【図3】



【図4】



【図5】





【図2】

